From Motion Processing to Autonomous Navigation

Julian Kolodko, Ljubo Vlacic, Liliane Peters[1]

Abstract – **This paper presents the overall structure of our work beginning with a brief introduction to motion processing algorithms moving onto the hardware architecture to be used for real time motion processing and finally introducing how motion information is used for autonomous navigation.**

I. INTRODUCTION

For humans, navigating in a complex, dynamic environment is second nature, however scientists are yet to design an autonomous robot that can reliably complete this task in an unstructured environment. Our aim is to bring this goal one step closer to reality by showing how current navigation approaches can be improved by explicitly incorporating real-time motion information into motion planning.

Using motion information explicitly is justified by research that indicates that motion is a fundamental visual dimension, much like colour and stereopsis [1]. In the brain motion information is fused with other information in a number of ways to allow humans to "see" and navigate in their environment. Our work focuses on the most obvious use motion perception; that is moving object segmentation and tracking. We do this based on the premise that by knowing where moving objects are going, a robot is better able to plan its path.

This presentation describes our environmental assumptions, choice of motion processing algorithms, proposed real-time hardware platform and proposed navigation scheme.

II. COMPUTATIONAL ASSUMPTIONS

Key to our work is the assumption that all motions occur on a smooth (but not flat) ground plain, and that all relevant (moving) objects touch this plain. This assumption immediately leads to two core simplifications: (i) we need only determine horizontal motion – vertical motion will remain small if all objects touch the ground plain. Indeed vertical motion reveals more about ground topology than object motion. (ii) the vertical extent of the input images need only be small if we are only looking for horizontal motion. Both these simplifications can lead to significant processing savings. In this work we assume a smooth ground plane to avoid additional problems introduced by camera shake.

Further, all "objects" in our work are items in the environment that are rigidly moving at a rate no faster than the maximum speed of our robots and are closer than a threshold determined by kinematic and sensor limitations. While there are further assumptions that appear in our work, they are in some sense implicit to the motion processing algorithm used so they are not discussed here.

III. MOTION PROCESSING

Motion Processing Algorithms. In the literature, a plethora of algorithms for motion processing have been proposed. These methods generally fall into 3 broad categories: *gradient based* where motion is derived from spatiotemporal image derivatives, *matching based* where some token is matched from one frame to the next and *frequency based* where frequency and/or phase information is used to determine motion [2].

Our work focuses on determining whether a robust 1D gradient based algorithm is more suitable than a robust 1D block-correlation based algorithm (a type of matching algorithm where image blocks are used as tokens). This assessment is being made in terms of (1) required computation, (2) consistency of computation across the image (consistent methods are more easily implemented in a parallel fashion), (3) ability to discriminate between multiple objects, (4) ability to determine object boundaries (5) range of measurable motions, (6) accuracy, (7) reliability under unfavorable conditions

Measurement of absolute velocity from an image sequence is impossible since we can not tell if an object is near by and moving slowly or far away and moving quickly. All we can say is that a particular image region is displaced some number of pixels between frames.

Non-robust implementations of the above algorithms show that gradient methods are better suited to small inter-frame displacements (<1.5 pixels) and can measure subpixel motion. Correlation methods can deal with a wider range of displacements, but are unable to resolve subpixel motion. In some respects this is beneficial because in real environments most subpixel motion is most likely "noise" motion. Future implementations utilizing robust statistical techniques will improve algorithm performance in suboptimal conditions and provide object localisation capability.

Approaching/Receding Motion. While it is possible to estimate the rate of approach of an object using visual information, this calculation is unreliable for objects that cover only a small portion of the image and is not directly supported by the above algorithms. To overcome this, our system will only use visual information to determine motion parallel to the camera surface. A laser range finder is being used to (a) measure rate of approach of objects and (b) confirm object boundaries.

Short Versus Long-Range Motion Estimation. The motion estimation approaches described above will at best provide an estimate of an objects motion and the location of the objects boundaries. Since these algorithms use

[1] Currently with Ericsson, Germany.

only two or three frames of data, they are referred to as "short range" algorithms.

The additional information provided over time by a sequence of images is utilised in "long range" algorithms, where motion estimates are made more accurate over time. This is achieved using a feedback system where earlier results are used as a basis for computing new results [3]. We employ two such mechanisms. Firstly, we implement motion estimation in an incremental fashion where previous results are minimally processed and fed back to the short-range algorithm. This must be implemented in a robust fashion so that erroneous results are not used as a basis for future computation. Secondly, we combine motion information with laser range information to track detected objects over time and determine a model for each objects motion. These motion models are used to generate a set of "predicted" motions, which are then fed back to the short-range motion estimation algorithm.

Testing. We are testing our algorithms in an offline simulation. This allows us to ensure our algorithms are correct without the additional problems (timing, architecture etc) related to implementation on a real-time hardware platform.

For simplicity we use MATLAB which is ideally suited to image processing applications. Each candidate algorithm is implemented then tested using a database of video sequences that are typical of the environment in which our robots operate. A database containing coordinated video and laser range information is under development to allow simulation of the overall system.

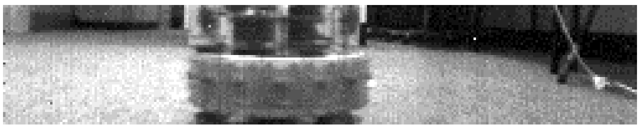

Figure 1. Example image from database

III. REAL TIME HARDWARE

Choice of platform. For motion information to be useful in a dynamic world, it must be extracted quickly relative to the velocity of objects in the environment so that navigation decisions can be made with up to date information. Unfortunately, motion processing is highly processor intensive due to the massive amount of information contained in a video data.

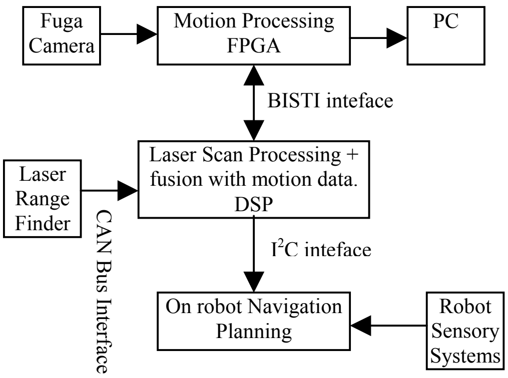

Figure 1. Global Architecture

To over come this, significant processing power is required in addition to clever and simple algorithms. To this end we have chosen the Signal Master platform coupled with a Gatesmaster add on board which provides us with a combination of Sharc DSP, 486 and Virtex FPGA processing platforms as well as a wealth of interface options. Video data comes from the compact Fuga 15D camera that does not require a frame grabber – rather it is accessed directly through a digital interface much like a typical RAM. Our laser range finder, produced by DLR, was chosen for its unique combination of small size, low power consumption and high scan rate.

Global Architecture. Figure 2 illustrates the global architecture of our proposed system and how it maps to hardware components and interfaces.

Motion processing, along with the requisite memory management functionality occurs in FPGA. For preliminary testing purposes this output can be fed to a PC for visualization. However in the final system motion processing results (that is, object location and motion) are passed to the DSP for further processing and fusion with laser range data.

The output from the DSP is a set of position and velocity estimates which pass both back to the FPGA for "long range" motion estimation and forward to the robot platform for navigation planning purposes.

V. AUTONOMOUS NAVIGATION

Real-time Dynamic Obstacle Reasoning. Autonomous navigation schemes are comprised of two integrated steps. Given the robots current location and its goal location, an overall, "long term" path plan is created. Then "short term planning" is performed by a static obstacle reasoning (SOR) unit whose goal is to avoid unexpected static obstacles and to keep the robot "on the road". This can often be an iterative scheme since the path plan may have to be reevaluated if the SOR stage detects the path is unfeasible.

Our intention is to incorporate a dynamic obstacle reasoning (DOR) stage into the navigation scheme. DOR will interact with SOR to generate dynamic changes to the robot's trajectory that will not lead to collisions in the short term. In order to prevent deadlock, these changes must also conform to some set of "rules of the road".

VI. CONCLUSIONS

This report has briefly presented the status of my Ph.D. work.

REFERENCES

[1] Nakayama K., "Biological Image Motion Processing: A Review", Vision Res, Vol 25, No 5, pp 625-660, 1985
[2] Beauchemin S. S., Barron J. L., "The Computation of Optical Flow", ACM Computing Surveys, Vol 27, No 3, pp 433-467, Sept 1995
[3] Black M. J., "Robust Incremental Optical Flow", Ph.D. Dissertation, Yale University, YaleU/CSD/RR #923, 1992